2024 15th International Conference on Information and Knowledge Technology (IKT)

KGLM-QA: A Knowledge Graph-Enhanced Large Language Model Approach to Question Answering

Alireza Akhavan Safaei
Department of Software Engineering
University of Isfahan
Isfahan, Iran
alirezaakhavansafaei@gmail.com

Pegah Saboori
Department of Software Engineering
University of Isfahan
Isfahan, Iran
p.saboori@eng.ui.ac.ir

Reza Ramezani
Department of Software Engineering
University of Isfahan
Isfahan, Iran
reza.ramezani@eng.ui.ac.ir

Mohammadali Nematbakhsh
Department of Software Engineering
University of Isfahan
Isfahan, Iran
nematbakhsh@eng.ui.ac.ir

***Abstract*— Large language models excel in various natural language processing tasks but often struggle with knowledge-intensive queries, particularly those involve rare entities or require precise factual information. This paper presents a novel framework that enhances capabilities of an LLM-based question answering system by incorporating structured knowledge from knowledge graphs. Our approach employs entity extraction, semantic similarity scoring, and adaptive graph exploration to efficiently navigate and extract relevant information from knowledge graphs. The core of the presented solution is a knowledge graph-enhanced language model process that iteratively refines subgraph exploration and answer generation, complemented by a fallback mechanism for robustness across diverse question types. Experiments on location-based questions from the Entity Questions dataset demonstrate significant improvements in the quality of responses. Using the Gemini 1.5 Flash model, our system achieved an accuracy increase from 36% to 71% for partially correct answers and from 22% to 69% for exactly correct answers, as evaluated by human assessors. This approach offers a promising direction for developing more reliable and accurate question answering systems, particularly for queries involving long-tail entities or specific factual knowledge.**

Keywords—Large Language Models, Knowledge graph, Question Answering, Retrieval augmented generation

I. INTRODUCTION

Todays, Large Language Models (LLMs) have revolutionized the field of Natural Language Processing, demonstrating impressive capabilities across a wide range of tasks [1]. Their ability to understand and generate human-like text has pushed the boundaries of what's possible in areas such as question answering, text summarization, and language translation. However, despite their remarkable performance, LLMs face significant challenges when it comes to knowledge-intensive tasks, particularly those requiring access to up-to-date, specialized, or less common information [2]. Recent research has demonstrated several challenges faced by large language models, especially in tasks involving factual knowledge [2].

LLMs exhibit limitations in encoding world knowledge, particularly when it comes to less popular or long-tail entities [2]. While scaling LLMs improves memorization of common facts, it fails to address their struggles with less prevalent knowledge. Retrieval augmentation has been shown to significantly improve performance in such cases, enhancing LLMs' ability to recall non-parametric knowledge when needed [3]. These limitations are especially concerning in domains requiring high accuracy, such as medicine, law, and scientific research.

Researchers have increasingly turned to knowledge graphs (KGs) to complement the limitations of LLMs, providing structured and explicit representations of knowledge [4]. KGs offer interpretability and accuracy in reasoning by representing facts in a structured way, like triples, which can help in knowledge-aware tasks such as question answering and reasoning [5]. However, there are several challenges in integrating KGs with LLMs, including filtering irrelevant information, supporting complex reasoning over multiple relationships, and effectively translating the structured knowledge of KGs into the free-text format that LLMs operate on [5].

This paper is an effort to leverage the strengths of both LLMs and knowledge graphs for enhanced question answering and reasoning tasks. To address these challenges, this paper introduces a framework that combines large language models with structured knowledge from knowledge graphs to enhance question answering accuracy.

Experiments demonstrate that our approach significantly enhances the performance of LLMs, particularly for queries that involve long-tail entities or require specific factual knowledge. The modular design of the framework allows for easy adaptation to various domains and models without incurring additional training costs, which as a result, represents a significant step towards more reliable and accurate question answering systems.

This work, we aim to contribute to the ongoing efforts to create more knowledgeable, accurate, and reliable AI systems for complex question answering tasks.

The remainder of this paper is organized as follows: the Background and Related Work reviews existing approaches in knowledge graph question answering and language model enhancement; the Proposed Approach outlines our entity-centric method, adaptive exploration, and iterative refinement; the Experimental Setup covers datasets, baseline models, and evaluation metrics; the Results and Discussion presents performance analysis and case studies; and the

Conclusion and Future Work summarizes findings and suggests future research directions.

II. Background and Related Work

The field of Question Answering has seen significant advancements in recent years, driven by the development of Large Language Models and the increasing availability of structured knowledge in the form of Knowledge Graphs. This section provides an overview of these key components and their roles in modern QA[1] systems.

Large Language Models are deep learning models trained on vast amounts of text data to understand and generate human-like text. These models, such as BERT[2] [6], and more recent models like GPT[3]-3 [7], have revolutionized natural language processing tasks, including question answering.

LLMs operate on the principle of self-attention and transformer architectures [8], allowing them to capture long-range dependencies in text and generate coherent responses. They have demonstrated remarkable capabilities in various NLP[4] tasks. Despite their impressive performance, LLMs face significant challenges, particularly in tasks requiring access to up-to-date, specialized, or less common information [2]. Their knowledge is inherently limited to the data they were trained on, which may result in inaccuracies or outdated information in responses [2]. Furthermore, LLMs do not have the capacity to remember all the information from their training data, given the sheer scale of the models and the vastness of the data they process. As a result, LLMs often bias towards more frequently occurring data, disproportionately favoring commonly seen entities and information, while struggling with long-tail entities.

Long-tail entities refer to less frequent or rare entities that are not as prominently represented in the training data [9]. Examples of long-tail entities include specific geographic locations, specialized technical terms, or obscure historical facts. These entities are harder for LLMs to handle because the models are more likely to memorize commonly occurring information, while long-tail entities receive less attention during training [2]. This makes LLMs prone to omitting or misrepresenting rare or specialized knowledge, further limiting their applicability in certain domains.

In addition to the challenges with long-tail entities, LLMs are also prone to hallucinations—a well-documented problem where the model generates plausible-sounding but factually incorrect or fabricated information. This occurs because LLMs, when they lack sufficient knowledge, still attempt to provide a response, even if it is inaccurate or completely fabricated [10]. Hallucination is particularly problematic in knowledge-intensive tasks, where precise and factual answers are essential.

To address these limitations, Knowledge Graphs offer a robust solution. KGs provide structured, machine-readable representations of factual information, typically in the form of triples (subject, predicate, object), which are explicitly designed to represent relationships between entities. By incorporating KGs into question answering systems, models can access verifiable and up-to-date information, significantly reducing the risk of hallucinations [4]. Additionally, KGs help overcome the issue of long-tail entities by providing direct access to specific facts about rare or specialized entities that LLMs may not have encountered frequently in their training data [5].

Recent years have seen significant advancements in combining language models with knowledge graphs for question answering tasks. Key studies have contributed to this field, focusing on methods for entity extraction, reasoning, and multi-hop QA.

In 2023, Jiang and colleagues introduced UniKGQA, a novel approach addressing the challenge of multi-hop question answering on knowledge graphs [12]. UniKGQA distinguishes itself by integrating the traditionally separate processes of retrieval and reasoning into a single, unified model. This approach utilizes a semantic matching module that employs a pre-trained language model to match the semantics of the question with relations in the knowledge graph. Additionally, a matched information propagation module propagates this matched information across the directed edges of the knowledge graph. For instance, when processing a question like "Who is the spouse of the nominee for the Nobel Prize in Literature?", UniKGQA first identifies key relations such as "nominee" and "spouse," matches these with knowledge graph relations, and then propagates information along paths from "Nobel Prize winner" to "nominee" to "spouse." This process creates a relevant subgraph from which the final answer entity is identified. By unifying retrieval and reasoning, UniKGQA demonstrates improved performance in complex, multi-hop question answering tasks.

Another significant contribution in 2023 came from Jiang et al. with the development of StructGPT, a framework designed to enhance the reasoning capabilities of large language models when working with structured data [13]. StructGPT adopts a read-then-reason approach, inspired by tool-augmented strategies for large language models. The framework employs specialized interfaces to efficiently gather relevant evidence from structured data sources such as tables, knowledge graphs, and databases. StructGPT's workflow involves an iterative process of information gathering and reasoning. When presented with a question, the model first uses specific interfaces to search for relevant information. It then converts this gathered data into a textual format that the language model can process. This process is repeated, with the model extracting increasingly detailed information and refining its reasoning with each iteration. For example, when answering a question about company earnings in 2021, StructGPT might first use knowledge graph interfaces to find general relationships about company earnings and CEOs, then use table interfaces to extract specific earnings data for the year in question. This iterative approach allows for more precise and comprehensive answers to complex queries involving structured data.

In 2022, Saxena and colleagues introduced KGT5, an innovative model that performs both link prediction in knowledge graphs and question answering using a sequence-to-sequence approach [14]. KGT5 redefines link prediction as a sequence-to-sequence task, enabling the use of an encoder-decoder transformer model. This approach uses textual representations of entities and relations to convert link prediction tasks into text-based questions. The model's

---

[1] Question Answering

[2] Bidirectional Encoder Representations from Transformers

[3] Generative Pre-trained Transformer

[4] Natural Language Processing

architecture is based on T5-small but is trained from scratch, first for link prediction and then fine-tuned for question answering while maintaining link prediction as a regularizing objective. KGT5 demonstrates superior performance in both tasks, significantly reducing the size of traditional link prediction models while delivering performance on par with or better than more complex models on large-scale question-answering benchmarks. A key advantage of KGT5 is its robustness in settings with incomplete knowledge graphs, showcasing its ability to handle missing data effectively.

Baek et al. (2023) developed KAPING, a zero-shot approach for knowledge graph question answering that enhances large language models with knowledge graph information [15]. KAPING's innovation lies in its ability to answer questions without the need for labeled samples or specific training. The approach transforms input questions into prompts using a specific command template and then augments these prompts with relevant information from knowledge graphs. For example, the question "What is the capital of France?" might be augmented to "Question: What is the capital of France? Info: France-capital-Paris. Answer:". This augmented prompt is then provided to the language model for answer generation. To enhance efficiency and relevance, KAPING employs semantic filtering to exclude irrelevant relations based on their similarity to the input question. This method demonstrates how the integration of knowledge graph information can significantly improve the accuracy of large language models in zero-shot question answering scenarios.

In a recent 2024 study, Gu and colleagues introduced the Knowledge Navigator framework, aimed at improving the reasoning capabilities of large language models, particularly for multi-hop question answering tasks [16]. Knowledge Navigator operates by combining large language models with knowledge retrieval from structured sources such as databases and knowledge graphs. This approach focuses on enhancing the model's ability to handle complex queries that require multiple steps of reasoning. By leveraging structured knowledge sources, Knowledge Navigator not only improves the accuracy and precision of answers to multi-step questions but also potentially enhances the explainability and traceability of the AI system's reasoning processes. This framework represents a significant step forward in bridging the gap between the broad knowledge capture of large language models and the structured, factual information contained in knowledge graphs and databases.

Previous approaches, such as UniKGQA [12] and StructGPT [13], have integrated large language models (LLMs) with knowledge graphs for question answering, but they often require extensive training or inefficiently handle complex graph structures. Our work introduces two key innovations in subgraph extraction that address these limitations. First, instead of training a new model to predict hops or asking LLMs to predict the number of hops directly, we utilize the LLM’s knowledge to generate SPARQL queries, which predict subgraph depth with greater interpretability. This approach avoids the need for further training, as required by models like Knowledge Navigator [16], and provides a clearer reasoning path.

Second, unlike previous methods where all links and entities are passed to the LLM for ranking simultaneously—an approach that struggles with the hub nodes typical of knowledge graphs—we propose an adaptive exploration mechanism. In our framework, entities and links are ranked individually using semantic similarity scoring, allowing for efficient ranking and targeted exploration. The model sorts entities by their scores, prunes irrelevant ones, and iteratively explores further, which significantly improves the efficiency and scalability when dealing with large amounts of data. These innovations make our approach more robust for handling complex queries while minimizing computational overhead.

III. Proposed Approach

In this section, we detail our novel approach to enhancing question answering systems by integrating large language models with structured knowledge from knowledge graphs. Our approach is designed to address the limitations of LLMs in handling knowledge-intensive tasks, particularly those requiring precise factual information or involving long tail entities. Figure 1 illustrates the overall workflow of our proposed framework.

As shown in Figure 1, the proposed framework consists of several key stages: Prompt Analysis, Knowledge Graph Analysis, and Retrieval-Augmented Generation (RAG). The Prompt Analysis stage includes entity extraction, SPARQL query generation, and hop prediction. The Knowledge Graph Analysis stage involves subgraph extraction from the knowledge graph and conversion of the relevant information into triples. Finally, the RAG stage combines the original prompt with the extracted knowledge graph information to generate a more accurate and informed answer. This framework combines the strengths of large language models with structured knowledge from knowledge graphs to enhance question answering accuracy and reliability. The system employs an entity-centric approach, leveraging advanced entity extraction and semantic similarity scoring to efficiently navigate knowledge graphs. In the following subsections, we elaborate on each of these components and their roles within the overall system.

A. *Entity Extraction and Linking*

Our system begins by identifying and extracting relevant entities from the input query. This step is crucial for mapping the query to the corresponding nodes in the knowledge graph. The entity extraction process forms the foundation of our entity-centric approach, enabling efficient navigation of the knowledge graph.

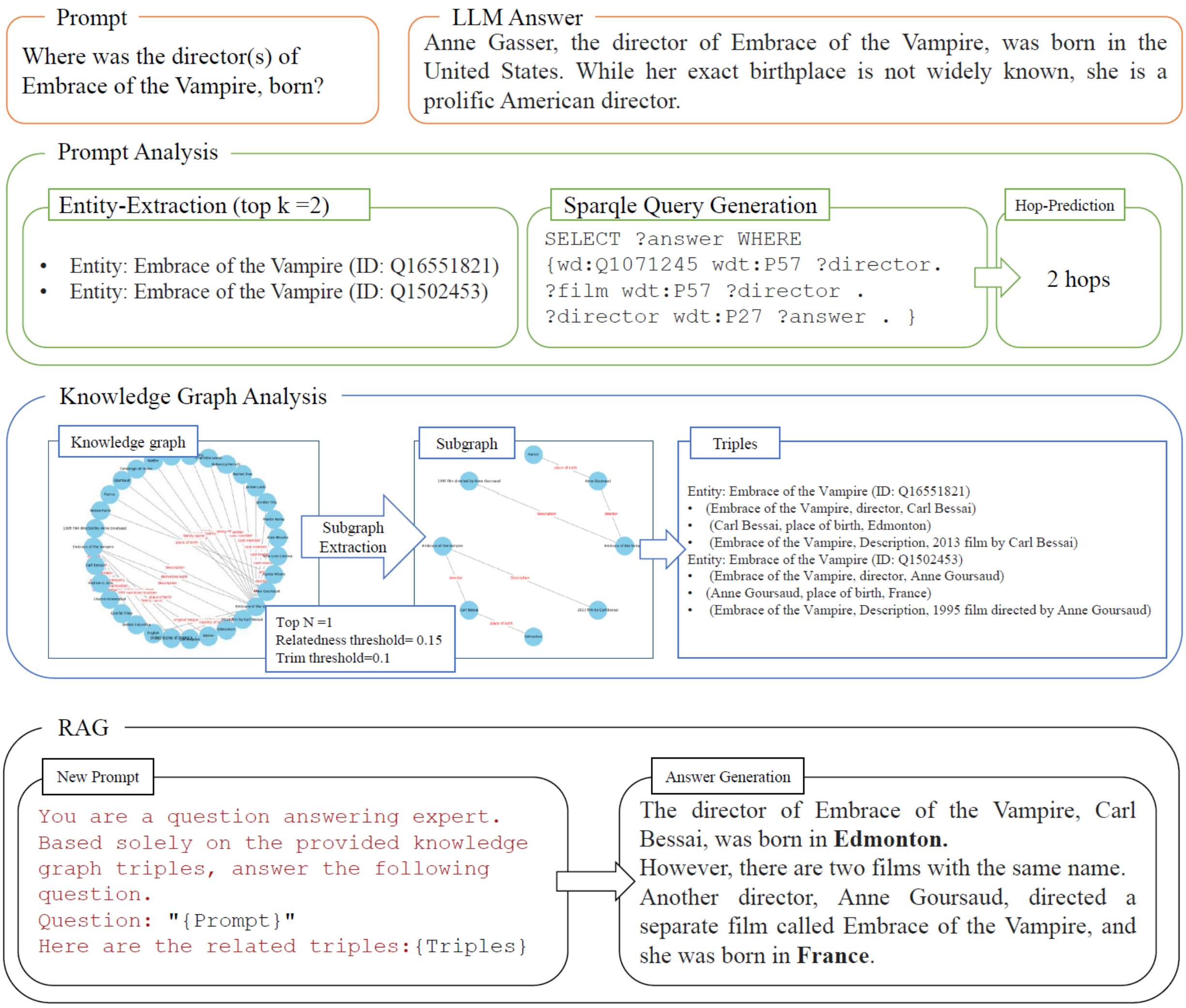

Fig. 1. Proposed Framework for KG-Enhanced LLM QA.

We propose a modular entity extraction approach that can be tailored to different contexts and domains. This flexibility allows for optimal performance across a wide range of question types. For general contexts, we leverage well-established, pre-trained transformer models that have demonstrated high accuracy in NER[5] tasks. And for domain-specific contexts, we recommend using fine-tuned models that have been specially adapted to recognize entities within particular fields (e.g., biomedical terms, legal entities, or technical jargon). This modular approach ensures that our system can be easily adapted to various domains without compromising on entity extraction accuracy.

Once entities are extracted from the query, the next crucial step is linking these entities to their corresponding entries in the knowledge graph. This process may yield multiple potential matches for each extracted entity. To handle this, we employ the following strategy:

1. For each extracted entity, we identify all potential matches in the knowledge graph.
2. We then use our semantic similarity scoring module (detailed in the next section) to rank these matches based on their relevance to the query context.
3. A configurable parameter 'k' determines how many of the top-ranked matches for each entity will be explored further. This parameter allows for fine-tuning the breadth of our knowledge graph exploration.

For example, consider a query about the publication date of a movie that has had several remakes. By adjusting the 'k' parameter, we can control whether our system explores just the most likely match (e.g., the original movie) or if it considers multiple versions of the movie in its exploration.

The output of this stage is a list of extracted entities from the query, each potentially linked to multiple entries in the knowledge graph, ranked by relevance. This forms the starting point for our subsequent knowledge graph exploration and question answering process.

B. Semantic Similarity Scoring

A crucial component of our framework is the semantic similarity scoring module, which plays a vital role in determining the relevance of knowledge graph entities/relations to the input question. This module enables our system to prioritize the most pertinent information from the knowledge graph, enhancing the accuracy and efficiency of the question answering process.

Our approach leverages advanced transformer models to compute semantic similarity between the input question and

[5] Named Entity Recognition

various elements from the knowledge graph. Specifically, we focus on two key comparisons:

1. Question-Entity Similarity: We compute the similarity between the input question and the descriptions of entities in the knowledge graph. This allows us to identify which entities are most relevant to the query.
2. Question-Relation Similarity: We also calculate the similarity between the question and the descriptions of relations in the knowledge graph. This helps in identifying which relationships between entities are most pertinent to answering the question.

This approach allows our system to effectively gauge the relevance of various knowledge graph elements to the input question, forming a crucial foundation for subsequent steps in our question answering pipeline.

C. Adaptive Knowledge Graph Exploration

Our framework incorporates an adaptive exploration mechanism that extracts relevant subgraphs from the knowledge base. This process is designed to dynamically adjust the depth of knowledge retrieval based on question complexity, ensuring efficient and targeted exploration of the knowledge graph.

The first step in our adaptive exploration is to predict the number of hops (or depth) needed to answer the question. We leverage the capabilities of a large language model to estimate this depth:

1. We construct a prompt that asks the language model to create a SPARQL query for the given question and entities.
2. The generated SPARQL query is then analyzed to count the number of hops required.
3. This hop count serves as our initial estimate for the depth of exploration needed.

The hop counting algorithm analyzes the structure of the SPARQL query, considering factors such as the number of triple patterns and their arrangement to determine the complexity of the query. Starting from the core entities identified in the previous steps, we perform an iterative exploration of the knowledge graph:

1. For each iteration (corresponding to a hop):
    a) We extract all nodes directly connected to the current set of entities.
    b) These connected nodes and their relationships are ranked based on their semantic similarity to the original question.
    c) Two pruning strategies are applied to filter the results:
        - A threshold (trim threshold) is applied to the semantic similarity score of the links.
        - Another threshold (relatedness threshold) is applied to the semantic similarity score of the target entities.

    d) From the remaining candidates, we select the top N entities, where N is a user-defined parameter.
2. This process is repeated for each hop, up to the maximum depth predicted.

By leveraging semantic similarity scores, our system adaptively focuses on the most relevant paths in the knowledge graph. The exploration process can be fine-tuned through several key parameters. The 'N' parameter determines how many highest-scoring entities to retain at each hop, allowing for control over the breadth of exploration. Two distinct thresholds play crucial roles during the exploration phase. A relation threshold is applied to the semantic similarity score of the links between entities. This ensures that only relationships deemed sufficiently relevant to the question are considered. And an entity threshold is applied to the semantic similarity score of the target entities. This strategy filters out entities that are not sufficiently related to the question, even if they are connected by a relevant relationship. These thresholds work in tandem to refine the exploration process, ensuring that both the relationships and the entities in the subgraph maintain a high degree of relevance to the original question.

D. Knowledge Graph-Enhanced Language Model Process

Once the relevant subgraph is extracted, our system leverages this structured knowledge to guide the large language model in generating accurate and contextually relevant answers. The integration process consists of the following steps:

1. Subgraph Representation: We represent the extracted subgraph as a set of triples, capturing the most pertinent entities and relationships identified during the exploration phase.
2. RAG Integration: These triples are injected into the prompt as a form of retrieval-augmented generation. This process provides the language model with specific, query-relevant factual information from the knowledge graph.
3. Answer Generation: The language model processes the query along with the added knowledge graph context. This allows the model to leverage both its pre-trained knowledge and the specific factual information from the knowledge graph to generate a response.

The knowledge graph-enhanced language model process forms the foundation of our system's ability to generate informed, accurate answers by combining structured knowledge with the natural language processing capabilities of large language models. This process represents a key step towards more reliable and accurate question answering systems, especially for knowledge-intensive tasks where precise factual information is crucial.

E. Fallback Mechanism

Our system incorporates a robust fallback mechanism to ensure consistent performance across a wide range of question types. This mechanism allows the system to seamlessly transition between using knowledge graph-augmented information and relying on the language model's

inherent knowledge when the retrieved information is insufficient or empty.

IV. Experimental Setup

To evaluate the effectiveness of our proposed approach, we conducted a series of experiments using a combination of state-of-the-art models and well-established knowledge bases. This section outlines the technical details of our implementation, including the computational environment, models employed, and specific parameters used in our experiments.

The experiments were conducted on Google Colab, utilizing the Gemini 1.5 Flash model via the Google Generative AI API. This model was used for key tasks, including question analysis, SPARQL query generation, and answer production. For named entity recognition and extraction, we employed SpaCy's transformer-based model, specifically the "en_core_web_trf" variant, which is built on RoBERTa and optimized by SpaCy for NER tasks. To compute semantic similarities between the input questions and knowledge graph components, we utilized the all-MiniLM-L6-v2 model from the Sentence Transformers family, which balances efficiency and accuracy for these tasks. Wikidata served as our primary knowledge base due to its extensive topic coverage and structured, machine-readable format.

The number of initial entities considered in the entity linking phase was limited to one, ensuring that only the most relevant option was selected for each extracted mention. During knowledge graph exploration, the system retained the two highest-scoring entities at each iteration, allowing for a balance between breadth and focus. To maintain relevance, any entities and relationships with semantic similarity scores below 0.1 were filtered out. Additionally, only initial entities with a similarity score of 0.1 or higher were considered relevant to the question for further exploration. These parameters were chosen to balance between the breadth of exploration and the relevance of retrieved information, ensuring efficient and focused knowledge graph traversal.

For our evaluation, we utilized a subset of the Entity Questions dataset, specifically focusing on location-based questions from the P131.test.json file [17]. This subset was chosen to assess our system's performance on queries requiring specific factual knowledge about geographical entities.

To assess the performance of our system, we employed human evaluators to judge the accuracy of the generated answers. Since the large language model's output may vary in format or use different names for the same entity, human evaluation was necessary to ensure accurate assessment by carefully reading and interpreting the answers. The evaluators determined whether each answer was correct, partially correct, or incorrect based on the given question and the known ground truth. We then calculated the accuracy score as the proportion of correct answers out of the total number of questions.

V. Results and Discussion

Our experiments demonstrate the effectiveness of the proposed Knowledge Graph-Enhanced Question Answering system compared to a standalone Large Language Model approach. The results show a significant improvement in accuracy when using our RAG approach.

TABLE I. THE ACCURACY SCORES FOR BOTH THE LLM-ONLY AND PROPOSED APPROACH

Approach	**Accuracy Score**	
	Partially Correct Answers	***Exactly Correct Answers***
LLM only	36%	22%
Proposed approach	71%	69%

Table 1 summarizes the accuracy scores for both the LLM-only approach and our RAG-based approach. These results indicate a substantial improvement in performance when using our RAG-based approach:

1. Partially Correct Answers: Our RAG-based system achieved a 71% accuracy rate for partially correct answers, compared to 36% for the LLM-only approach. This represents a 97.2% relative improvement.
2. Exactly Correct Answers: The improvement is even more pronounced for exactly correct answers, with our RAG-based system achieving 69% accuracy compared to 22% for the LLM-only approach. This represents a 213.6% relative improvement.

To illustrate the difference in answer quality between the two approaches, consider the following example:

Question: Where is Toccoa located?

- LLM-only answer: "[Georgia]", which is its state.
- RAG-based answer: "[Stephens County]"
- Ground truth: "Stephens County"

In this case, the LLM-only approach provides a partially correct answer by identifying the state (Georgia) but fails to provide the specific county. Our RAG-based system, however, correctly identifies the exact location (Stephens County) as given in the ground truth.

This example demonstrates how our system's integration of knowledge graph information allows for more precise and accurate answers, particularly for questions requiring specific factual knowledge.

The significant improvement in accuracy demonstrates the effectiveness of our Knowledge Graph-Enhanced Question Answering system. By leveraging structured knowledge from Wikidata and combining it with the natural language understanding capabilities of large language models, our system is able to provide more accurate and specific answers, particularly for questions involving geographic entities.

The high accuracy for exactly correct answers (69%) in our RAG-based approach is particularly noteworthy. This suggests that the system is not only able to understand the question and provide relevant information but can also pinpoint the exact answer required, which is crucial for many real-world applications.

The improvement in partially correct answers (from 36% to 71%) indicates that even when the system doesn't provide the exact ground truth, it still offers relevant and useful information more frequently than the LLM-only approach.

To further illustrate the advantages of our approach, consider the following example:

Question: Where is Bala Shekar Kesh located?

- LLM-only answer: "I do not have access to real-time information, including specific locations like 'Bala Shekar Kesh'. To find this information, I recommend you try a search engine like Google or a map service like Google Maps or Apple Maps."
- RAG-based answer: "[Gilan Province] Bala Shekar Kesh is located in Gilan Province, Iran."

In this case, the LLM-only approach fails to provide any specific information about the location, instead suggesting external resources. Our RAG-based system, however, correctly identifies the location as Gilan Province in Iran.

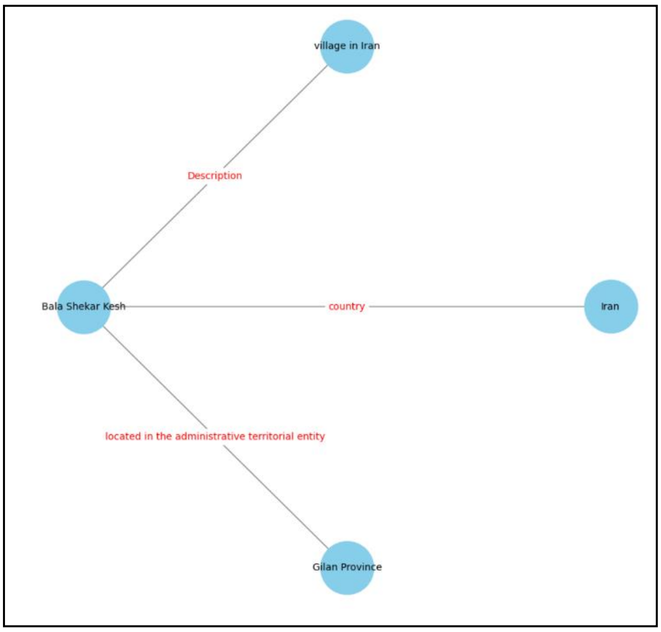

Fig. 2. Example of retrieved subgraph.

Figure 2 illustrates the knowledge graph subgraph retrieved for this question, which includes the following information:

Entity: Bala Shekar Kesh (ID: Q5792393)

(Bala Shekar Kesh, located in the administrative territorial entity, Gilan Province)

(Bala Shekar Kesh, country, Iran)

(Bala Shekar Kesh, Description, village in Iran)

This example demonstrates how our system effectively leverages the structured information from the knowledge graph to provide accurate answers, even for less common or more specific geographic entities that may not be well-represented in the LLM's training data.

These results validate our hypothesis that integrating knowledge graph information with large language models can significantly enhance question answering performance, especially for tasks requiring specific factual knowledge. The system's ability to provide accurate information about lesser-known locations like Bala Shekar Kesh underscores its potential for handling a wide range of geographical queries, including those involving less prominent or more specialized locations.

Furthermore, this example highlights the system's capability to not only provide the direct answer (Gilan Province) but also to offer additional relevant context (that it's in Iran and is a village). This additional information demonstrates the depth of understanding our system can achieve by combining knowledge graph data with language model capabilities.

VI. Conclusion and Future Work

This paper presents a novel approach to enhancing question answering systems by integrating large language models with structured knowledge from knowledge graphs. Our proposed framework addresses the limitations of LLMs in handling knowledge-intensive tasks, particularly those requiring precise factual information or involving long-tail entities. The experimental results demonstrate significant advantages of our Knowledge Graph-Enhanced Question Answering system. Most notably, our RAG-based approach achieved a 69% accuracy rate for exactly correct answers, compared to 22% for the LLM-only approach—a substantial 213.6% relative improvement. The system exhibited particular strength in handling questions about lesser-known or more specialized geographical entities, showcasing its potential for a wide range of knowledge-intensive tasks. This capability was evident in examples such as accurately locating Bala Shekar Kesh, where the LLM-only approach failed to provide specific information. By effectively combining structured data from knowledge graphs with the natural language processing capabilities of LLMs, our system consistently provided more accurate and contextually relevant answers. This integration of knowledge sources not only improved the accuracy of responses but also enhanced the depth and specificity of the information provided, as seen in the additional context offered for geographical queries. These results underscore the potential of our approach in advancing the field of question answering, particularly for tasks requiring access to specific, factual knowledge that may be beyond the training data of current large language models.

These results validate our hypothesis that integrating knowledge graph information with large language models can significantly enhance question answering performance, especially for tasks requiring specific factual knowledge.

However, our study has revealed several challenges and areas for improvement:

1. Scalability in large knowledge graphs: When working with extensive knowledge bases like Wikidata, we encountered the challenge of hub nodes—entities with a vast number of connections. These hubs can cause the scale of the subgraph to grow exponentially within just a few hops, making efficient pruning crucial for effective exploration.
2. Hop count prediction: Accurately predicting the number of hops needed for exploration proved challenging, as the LLM may not fully understand the structure of the knowledge graph. This can lead to suboptimal exploration depths.
3. Query relation utilization: Our current approach could be enhanced by better leveraging the relations

expressed in the query itself, which could provide valuable guidance for knowledge graph exploration.

4. RAG decision-making: Determining when to use the RAG approach versus relying solely on the LLM remains an area for optimization.

5. Natural language conversion: The potential benefits of converting knowledge graph triples into natural language format to better align with LLM training data have yet to be fully explored.

In conclusion, our Knowledge Graph-Enhanced Question Answering system represents a significant step forward in combining the strengths of large language models and structured knowledge bases. While we have demonstrated substantial improvements in accuracy and capability, our work has also illuminated several critical challenges in this field. As we continue to refine and expand this approach, addressing these challenges and exploring the identified areas for future work, we anticipate further advancements in the accuracy, reliability, and versatility of question answering systems. These improvements will pave the way for more intelligent and capable AI assistants across various applications and industries, bringing us closer to the goal of creating truly knowledgeable and context-aware artificial intelligence.

REFERENCES

[1] H. Naveed et al., "A comprehensive overview of large language models," arXiv preprint arXiv:2307.06435, 2023.

[2] A. Mallen et al., "When not to trust language models: Investigating effectiveness of parametric and non-parametric memories," arXiv preprint arXiv:2212.10511, 2022.

[3] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," Advances in Neural Information Processing Systems, vol. 33, pp. 9459-9474, 2020.

[4] J. Z. Pan et al., "Large language models and knowledge graphs: Opportunities and challenges," arXiv preprint arXiv:2308.06374, 2023.

[5] S. Pan et al., "Unifying large language models and knowledge graphs: A roadmap," IEEE Transactions on Knowledge and Data Engineering, 2024.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805, 2018.

[7] T. Brown et al., "Language Models are Few-Shot Learners," arXiv preprint arXiv:2005.14165, 2020.

[8] A. Vaswani et al., "Attention is All You Need," Advances in Neural Information Processing Systems, 2017.

[9] N. Kandpal et al., "Large language models struggle to learn long-tail knowledge," in International Conference on Machine Learning, 2023, pp. 15696-15707.

[10] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv preprint arXiv:2312.10997, 2023.

[11] A. Creswell, M. Shanahan, and I. Higgins, "Selection-inference: Exploiting large language models for interpretable logical reasoning," arXiv preprint arXiv:2205.09712, 2022.

[12] J. Jiang, K. Zhou, W. X. Zhao, and J. R. Wen, "Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph," arXiv preprint arXiv:2212.00959, 2022.

[13] J. Jiang et al., "Structgpt: A general framework for large language model to reason over structured data," arXiv preprint arXiv:2305.09645, 2023.

[14] A. Saxena, A. Kochsiek, and R. Gemulla, "Sequence-to-sequence knowledge graph completion and question answering," arXiv preprint arXiv:2203.10321, 2022.

[15] J. Baek, A. F. Aji, and A. Saffari, "Knowledge-augmented language model prompting for zero-shot knowledge graph question answering," arXiv preprint arXiv:2306.04136, 2023.

[16] T. Guo et al., "Knowledgenavigator: Leveraging large language models for enhanced reasoning over knowledge graph," Complex & Intelligent Systems, pp. 1-14, 2024.

[17] C. Sciavolino, Z. Zhong, J. Lee, and D. Chen, "Simple entity-centric questions challenge dense retrievers," arXiv preprint arXiv:2109.08535, 2021.